GRAPHS & DIGRAPHS

Second Edition

Gary Chartrand

Linda Lesniak

GRAPHS & DIGRAPHS

Second Edition

Gary Chartrand *Western Michigan University*

Linda Lesniak *Drew University*

Wadsworth & Brooks/Cole Advanced Books & Software
Monterey, California
A division of Wadsworth, Inc.

To our friend and colleague

Farrokh Saba

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Chartrand, Gary.
Graphs & digraphs.

(The Wadsworth & Brooks/Cole mathematics series)
Rev. ed. of: Graphs & digraphs / Mehdi Behzad, Gary Chartrand, Linda Lesniak-Foster. c1979.
Bibliography: p. 336
Includes index.
1. Graph theory. 2. Directed graphs. I. Lesniak, Linda. Graphs & digraphs.
III. Title. IV. Title: Graphs and digraphs.
V. Series.
QA166.C4525 1986 511'.5 86-1519

ISBN 0-534-06324-1

Sponsoring Editor: John Kimmel
Editorial Assistant: Maria Alsadi
Production: Greg Hubit Bookworks
Typesetting: Permanent Typesetting and Printing Co., Ltd.
Printing and Binding: Maple-Vail Book Manufacturing Group

Preface

The exciting and rapidly growing area of graph theory is rich in theoretical results as well as applications to real-world problems. In this edition of *Graphs & Digraphs*, as in the first, our major emphasis is on the theoretical aspects of graph theory, and we have included what we believe to be the most interesting and important results in the field. In addition, however, we have introduced the reader to types of problems that can be modeled by graphs and we have indicated efficient algorithms for their solutions. In keeping with our belief that a background emphasizing theory and proof techniques is indispensable for the student of graph theory, we have included careful proofs that the algorithms do, in fact, accomplish what they claim. Exercises reflecting the addition of these algorithms as well as a substantial number of new exercises have been added.

A second major change in this edition is the integration of graph and digraph theory. The material on digraph theory, self-contained in the first edition, is now developed parallel to that of (undirected) graphs. This allows, for example, the max-flow min-cut theorem to be introduced early in the text and then used to establish results on connectivity and matching.

This text is intended for an introductory sequence in graph theory at the senior or beginning graduate level. However, a one-semester course could easily be designed by selecting those topics of major importance and interest to the students involved. To facilitate such a choice in this edition, we have judiciously chosen a number of topics to introduce and develop in the exercises rather than in the text itself. Three topics that are introduced early in the text can be omitted with little effect on the material that follows, namely Section 2.4 on the Reconstruction Problem, Section 3.2 on n-ary trees, and Sections 4.4–4.6 on embedding graphs on surfaces of positive genus.

It is a pleasure to thank a number of individuals who assisted us with this edition in a variety of ways. The discussions we had with Farhad Shahrokhi on graph algorithms were very useful to us, and we are most appreciative of the time and effort he spent on our behalf. We are grateful for the suggestions made by Garry Johns, Paresh J. Malde, Ortrud R. Oellermann, Robert Rieper, and Farrokh Saba. The advice given to us by reviewers of this edition was very helpful; we are delighted to thank Ruth A. Bari, Ralph Faudree, Ronald J. Gould, Jerrold R. Griggs, F. C. Holroyd, Gary T. Myers, and Richard D. Ringeisen. Our gratitude goes to Margo Johnson for her consistently excellent typing. Finally, we thank the staff of Wadsworth & Brooks/Cole Advanced Books & Software, particularly John Kimmel, for their interest in and assistance with this edition.

Gary Chartrand
Linda Lesniak

Contents

1 **Graphs and Digraphs** 1

1.1 Graphs 1
1.2 Digraphs 14
1.3 Degree Sequences 19

2 **Connected Graphs and Digraphs** 26

2.1 Paths and Cycles 26
2.2 Cut-Vertices, Bridges, and Blocks 45
2.3 Eulerian Graphs and Digraphs 52
2.4 An Unsolved Problem in Graph Theory: The Reconstruction Problem 62

3 **Trees** 66

3.1 Elementary Properties of Trees 66
3.2 *n*-Ary Trees 77
3.3 Decomposition of Graphs into Acyclic Subgraphs 83

4 **Graph Embeddings** 87

4.1 Euler's Formula 87
4.2 Characterizations of Planar Graphs 94

4.3 Nonplanar Graphs 107
4.4 The Genus of a Graph 117
4.5 2-Cell Embeddings of Graphs 125
4.6 The Maximum Genus of a Graph 141

5 Connectivity and Networks 152

5.1 n-Connected and n-Edge-Connected Graphs 153
5.2 Menger's Theorem 157
5.3 Networks 164
5.4 The Max-Flow Min-Cut Theorem 169
5.5 Applications of the Max-Flow Min-Cut Theorem 177

6 Hamiltonian Graphs and Digraphs 182

6.1 Hamiltonian Graphs 182
6.2 Hamiltonian Planar Graphs 196
6.3 Hamiltonian Digraphs 201

7 Oriented Graphs 207

7.1 Robbins' Theorem 207
7.2 Tournaments 209
7.3 Hamiltonian Tournaments 213
7.4 Score Sequences of Tournaments 215

8 Factors and Factorizations 225

8.1 Matchings 225
8.2 Factorizations 229
8.3 Decompositions 239
8.4 Coverings 243

9 Graphs and Groups 250

9.1 The Group and Edge-Group of a Graph 250
9.2 Cayley Color Graphs 256
9.3 Line Graphs 261

10 Graph Colorings 271

10.1 Vertex Colorings 271
10.2 Edge Colorings 286
10.3 Map Colorings 294

11 Extremal Graph Theory and Ramsey Theory 301

11.1 Extremal Graph Theory 301
11.2 Ramsey Numbers 306
11.3 Generalized Ramsey Theory 312

12 Enumeration of Graphs and Digraphs 319

12.1 Pólya's Theorem 319
12.2 Graphical and Digraphical Enumeration 329

Glossary of Symbols 333

Bibliography 337

Index 355

Chapter One

Graphs and Digraphs

In many disciplines we are faced with situations in which we want to find out how or whether a finite number of objects are related. If the relation is symmetric, we can model the situation by a graph. More generally, we can model the structure by a digraph. Hence, graphs and digraphs occur naturally and often. We begin our study with these two basic concepts.

1.1 Graphs

Many situations and structures give rise to graphs. Before we offer a precise definition of a graph, we present a few examples.

Assume that a California-based airline services several cities within California as well as Reno and Las Vegas, Nevada. These cities are indicated on the map shown in Figure 1.1(a).

This airline has several direct routes between certain pairs of these cities; the flying patterns are illustrated in Figure 1.1(b). The diagram in Figure 1.2(a) representing the cities serviced and the flying routes is a graph.

At times it is convenient to include additional information in a graph. For example, we might want to know the cost of each direct route. These costs (or weights) can be assigned to the edges of Figure 1.2(a), producing the network of Figure 1.2(b), where the labels a, b, and so on represent the costs.

By inspecting Figure 1.2, we can answer questions such as whether one can fly from San Diego to Reno and, if so, which route is least expensive. Of course, as graphs become more complex, solutions by inspection are no longer

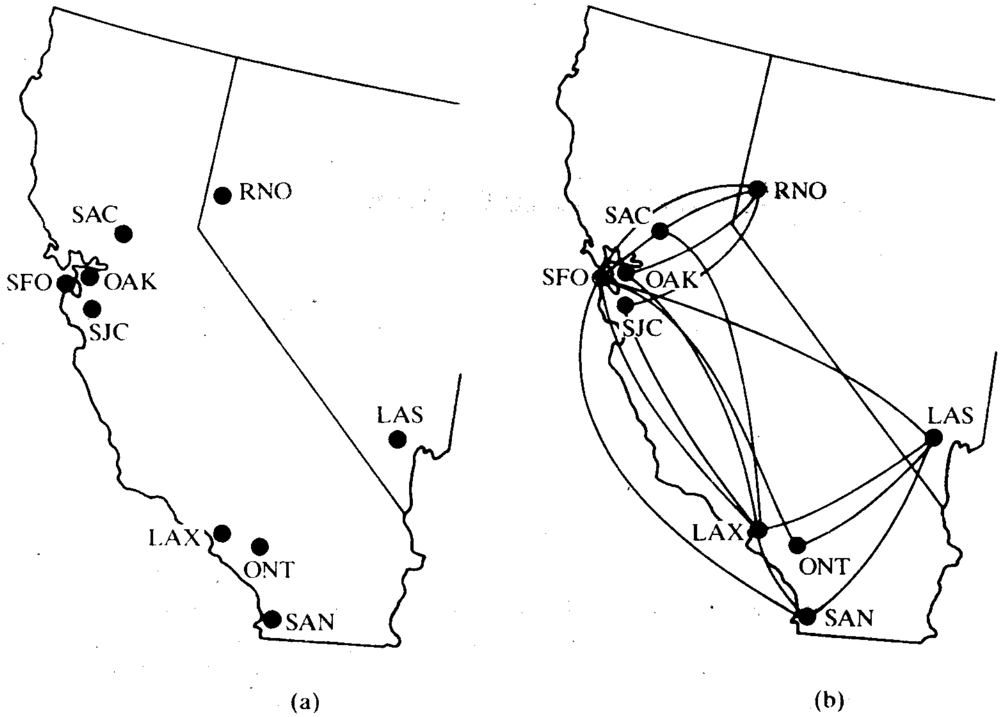


Figure 1.1 *Airline routes*

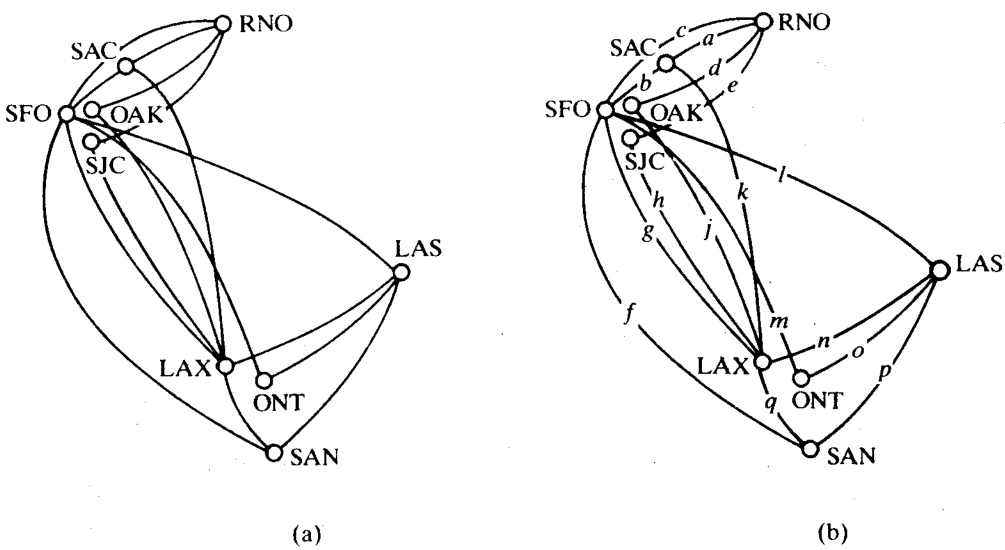


Figure 1.2 *A graph and a network*

feasible. In Chapter 2 we will discuss an efficient means of finding a "shortest path" in any graph.

As a second example, assume that a business is expanding and plans to add several new positions; namely, a draftsman, an engineer, a computer programmer, a data analyst, and an assistant personnel manager. Seven individuals apply for these five positions, some of whom have the qualifications for two or more of the positions. This situation can be represented by the graph shown in Figure 1.3, where five points (denoted DRA, ENG, PRO, ANL, and PER) are used to indicate the positions and seven points (1, 2, ..., 7) are used to indicate the applicants. Each point on the top of the graph represents an applicant, and each point on the bottom represents a position. A line is drawn between two points if the person is qualified for that position. A question that might be of interest is whether there are five individuals, from among the seven, who can be hired to fill all five positions. In graph theoretic terms, we are asking whether the set of jobs can be "matched" to a subset of the applicants. An algorithm that answers such questions will be discussed in Chapter 8.

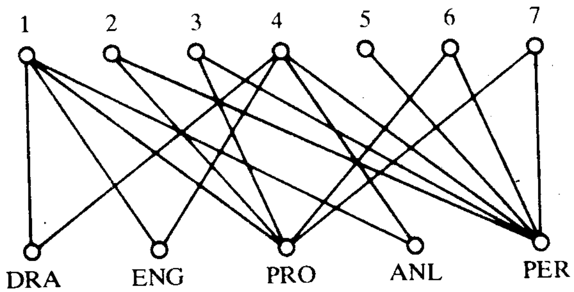


Figure 1.3 *A graph of jobs and applicants*

As a last example, let us suppose that eight experimental chemicals (A, B, ..., H) are to be stored in large (expensive) storage bins. Some chemicals have the potential to interact with each other and, consequently, should not be stored in the same bin. This situation is illustrated in the graph of Figure 1.4, where each chemical is represented by a point and two points are joined by a line if the corresponding chemicals should not be stored together. We might ask: What is the least number of storage bins that are needed to store all eight chemicals? This type of question is of particular interest to graph theorists. At the present, the only known algorithms to solve problems of this type are very inefficient, and many mathematicians believe that no efficient solution exists. We will see in Chapter 10 an example of an efficient "heuristic" algorithm for this problem; that is, an algorithm that describes a *small*, but not the *least*, number of bins that will suffice.

Each of the examples discussed so far was based on a collection of objects (cities, people, jobs, chemicals), and relationships between certain pairs. These

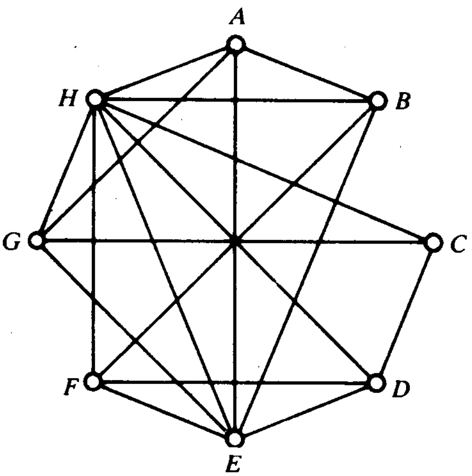


Figure 1.4 *A chemical interaction graph*

ideas are easily abstracted to produce the concept of a graph.

A *graph G* is a finite nonempty set of objects called *vertices* (the singular is *vertex*) together with a (possibly empty) set of unordered pairs of distinct vertices of G called *edges*. The *vertex set* of G is denoted by $V(G)$, while the *edge set* is denoted by $E(G)$.

The edge $e = \{u, v\}$ is said to *join* the vertices u and v. If $e = \{u, v\}$ is an edge of a graph G, then u and v are *adjacent vertices*, while u and e are *incident*, as are v and e. Furthermore, if e_1 and e_2 are distinct edges of G incident with a common vertex, then e_1 and e_2 are *adjacent edges*. It is convenient to henceforth denote an edge by uv or vu rather than by $\{u, v\}$.

The cardinality of the vertex set of a graph G is called the *order* of G and is denoted by $p(G)$, or more simply, p, while the cardinality of its edge set is the *size* of G and is denoted by $q(G)$ or q. A (p, q) *graph* has order p and size q.

It is customary to define or describe a graph by means of a diagram in which each vertex is represented by a point (which we draw as a small circle) and each edge $e = uv$ is represented by a line segment or curve joining the points corresponding to u and v.

A graph G with vertex set $V(G) = \{v_1, v_2, \ldots, v_p\}$ can also be described by means of matrices. One such matrix is the $p \times p$ *adjacency matrix* $A(G) = [a_{ij}]$, where

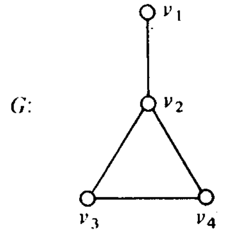
$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E(G) \\ 0 & \text{if } v_i v_j \notin E(G) \end{cases}$$

Thus, the adjacency matrix of a graph G is a symmetric $(0, 1)$ matrix having zero entries along the main diagonal.

For example, a graph G is defined by the sets

$$V(G) = \{v_1, v_2, v_3, v_4\} \quad \text{and} \quad E(G) = \{v_1v_2, v_2v_3, v_2v_4, v_3v_4\}.$$

A diagram of this graph and its adjacency matrix are shown in Figure 1.5.



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 1.5 *A graph and its adjacency matrix*

The adjacency matrix representation of a graph is often convenient if one intends to use a computer to obtain some information or solve a problem concerning the graph. On the other hand, an adjacency matrix contains a great deal of extraneous data—often many 0's and twice as many 1's as needed. This unsatisfactory characteristic of the adjacency matrix is often alleviated by inputting the graph in a variety of other manners. For example, one could input the edge set and the order, or one could input adjacency arrays, where the vertices adjacent to a given vertex are listed. There are several other possibilities. The manner in which a graph is input normally depends on the problem to be solved and affects the algorithm and method chosen to solve the problem.

Two graphs often have the same structure, differing only in the way their vertices and edges are labeled or in the way they are drawn. To make this idea more exact, we introduce the concept of isomorphism. A graph G_1 is *isomorphic* to a graph G_2 if there exists a one-to-one mapping ϕ, called an *isomorphism*, from $V(G_1)$ onto $V(G_2)$ such that ϕ preserves adjacency; that is, $uv \in E(G_1)$ if and only if $\phi u \phi v \in E(G_2)$. It is easy to see that "is isomorphic to" is an equivalence relation on graphs; hence, this relation divides the collection of all graphs into equivalence classes, two graphs being *nonisomorphic* if they are in different equivalence classes. If G_1 is isomorphic to G_2, then we say G_1 and G_2 are *isomorphic* and write $G_1 \cong G_2$.

Each of the graphs G_i, $i = 1, 2, 3$, of Figure 1.6 is a $(6, 9)$ graph. Here, G_1 and G_2 are isomorphic. For example, the mapping $\phi: V(G_1) \rightarrow V(G_2)$ defined by

$$\phi v_1 = v_1, \quad \phi v_2 = v_3, \quad \phi v_3 = v_5, \quad \phi v_4 = v_2, \quad \phi v_5 = v_4, \quad \phi v_6 = v_6$$

is an isomorphism. On the other hand, $G_1 \not\cong G_3$ since, for example, G_3 contains three pairwise adjacent vertices whereas G_1 does not. Of course, $G_2 \not\cong G_3$.

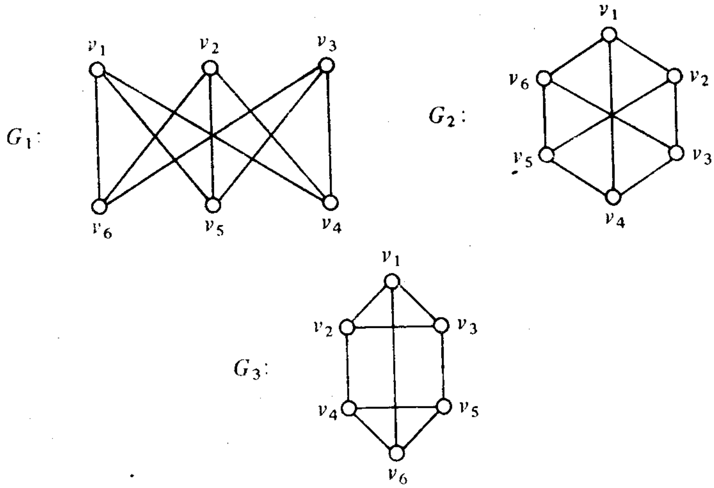


Figure 1.6 *Isomorphic and nonisomorphic graphs*

If G is a (p, q) graph, then $p \geq 1$ and $0 \leq q \leq \binom{p}{2} = p(p-1)/2$. There is only one (1, 0) graph (up to isomorphism), and this is referred to as the *trivial graph*. A *nontrivial graph* then has $p \geq 2$.

Two graphs G_1 and G_2 are *identical*, denoted $G_1 = G_2$, if $V(G_1) = V(G_2)$ and $E(G_1) = E(G_2)$. Clearly, two graphs may be isomorphic yet not identical. The graphs G_1 and G_2 of Figure 1.6 are not identical (even though $V(G_1) = V(G_2)$ and $G_1 \cong G_2$) since, for example, $v_1v_5 \in E(G_1)$ and $v_1v_5 \notin E(G_2)$.

All 20 nonidentical graphs of order 4 and size 3, having vertex set $\{1, 2, 3, 4\}$, are shown in Figure 1.7. Among these graphs, there are only three

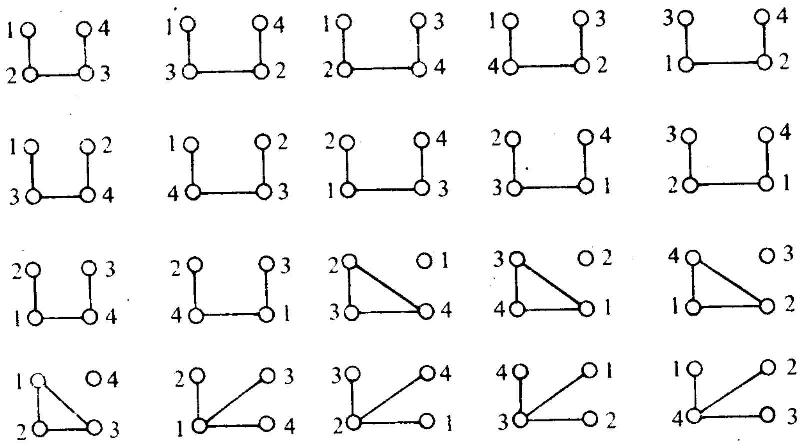


Figure 1.7 *The nonidentical* (4, 3) *graphs having vertex set* $\{1, 2, 3, 4\}$

nonisomorphic classes of graphs. The total number of nonidentical graphs having vertex set $\{1, 2, 3, 4\}$ is 64; in fact, the total number of nonidentical graphs of order p with the same vertex set V is $2^{p(p-1)/2}$. This is obvious for $p = 1$. If $p \geq 2$ and G is a graph with vertex set $V(G)$, then for each pair u, v of distinct vertices, there are two possibilities depending on whether uv is or is not an edge of G. Since there are $p(p-1)/2$ distinct pairs of vertices, there are $2^{p(p-1)/2}$ such nonidentical graphs G.

With the exception of the order and the size, the numbers that one encounters most frequently in the study of graphs are the degrees of its vertices. The *degree of a vertex* v in a graph G is the number of edges of G incident with v. The degree of a vertex v in G is denoted $\deg_G v$ or simply $\deg v$ if G is clear from the context. A vertex is called *odd* or *even* depending on whether its degree is odd or even. A vertex of degree 0 in G is called an *isolated vertex* and a vertex of degree 1 is an *end-vertex* of G. In Figure 1.8, a graph G is shown together with the degrees of its vertices.

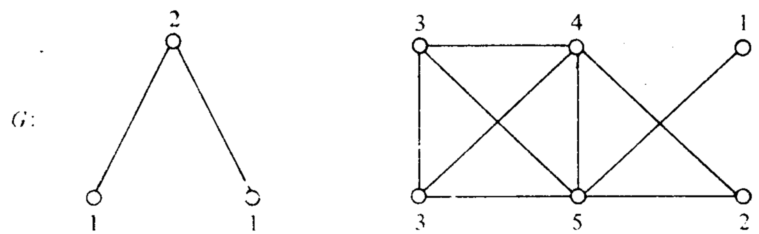


Figure 1.8 *The degrees of the vertices of a graph*

Observe that for the graph G in Figure 1.8, $p = 9$ and $q = 11$, while the sum of the degrees of its nine vertices is 22. The fact that this last number equals $2q$ for the graph G is not merely a coincidence. Every edge is incident with two vertices; hence, when the degrees of the vertices are summed, each edge is counted twice. We state this as our first theorem, which, not so coincidentally, is sometimes called "The First Theorem of Graph Theory".

Theorem 1.1 *Let G be a (p, q) graph where $V(G) = \{v_1, v_2, \ldots, v_p\}$. Then*

$$\sum_{i=1}^{p} \deg v_i = 2q.$$

This result has an interesting consequence.

Corollary 1.1 *In any graph, there is an even number of odd vertices.*

Proof Let G be a graph of size q. Also, let W be the set of odd vertices of G and let U be the set of even vertices of G. By Theorem 1.1,

$$\sum_{v \in V(G)} \deg v = \sum_{v \in W} \deg v + \sum_{v \in U} \deg v = 2q.$$

Certainly, $\Sigma_{v \in U} \deg v$ is even; hence $\Sigma_{v \in W} \deg v$ is even, implying that $|W|$ is even and thereby proving the corollary. ■

Frequently, a graph under study is contained within some larger graph also being investigated. We consider several instances of this now. A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$; in such a case, we also say that G is a *supergraph* of H. If G and H are graphs, not all of whose vertices are labeled, then H is also considered to be a subgraph of G if any unlabeled vertices can be labeled so that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If H is a subgraph of G, then we write $H \subset G$.

The simplest type of subgraph of a graph G is that obtained by deleting a vertex or edge. If $v \in V(G)$ and $|V(G)| \geq 2$, then $G - v$ denotes the subgraph with vertex set $V(G) - \{v\}$ and whose edges are all those of G not incident with v; if $e \in E(G)$, then $G - e$ is the subgraph having vertex set $V(G)$ and edge set $E(G) - \{e\}$. The deletion of a set of vertices or set of edges is defined analogously. These concepts are illustrated in Figure 1.9.

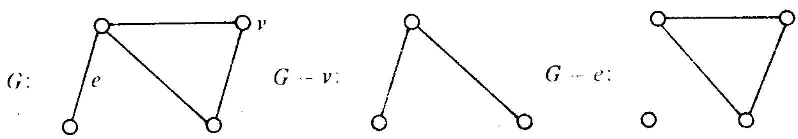


Figure 1.9 *The deletion of an element of a graph*

If u and v are nonadjacent vertices of a graph G, then $G + f$, where $f = uv$, denotes the graph with vertex set $V(G)$ and edge set $E(G) \cup \{f\}$. Clearly, $G \subset G + f$.

We have seen that $G - e$ has the same vertex set as G and that G has the same vertex set as $G + f$. Whenever a subgraph H of a graph G has the same order as that of G, then H is called a *spanning subgraph* of G.

Among the most important subgraphs we shall encounter are the "induced subgraphs". If U is a nonempty subset of the vertex set $V(G)$ of a graph G, then the subgraph $\langle U \rangle$ of G *induced* by U is the graph having vertex set U and whose edge set consists of those edges of G incident with two elements of U. A subgraph H of G is called *vertex-induced* or *induced*, denoted $H < G$, if $H \cong \langle U \rangle$ for some subset U of $V(G)$. Similarly, if F is a nonempty subset of $E(G)$, then the subgraph $\langle F \rangle$ *induced* by F is the graph whose vertex set consists of those vertices of G incident with at least one edge of F and whose edge set is F. A subgraph H of G is *edge-induced* if $H \cong \langle F \rangle$ for some subset F of $E(G)$. It is a simple consequence of the definitions that every induced subgraph of a graph G can be obtained by removing vertices from G

while every subgraph of G can be obtained by deleting vertices and edges. These concepts are illustrated in Figure 1.10 for the graph G, where

$$V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6\}, \quad U = \{v_1, v_2, v_5\}, \quad \text{and} \quad F = \{v_1v_4, v_2v_5\}.$$

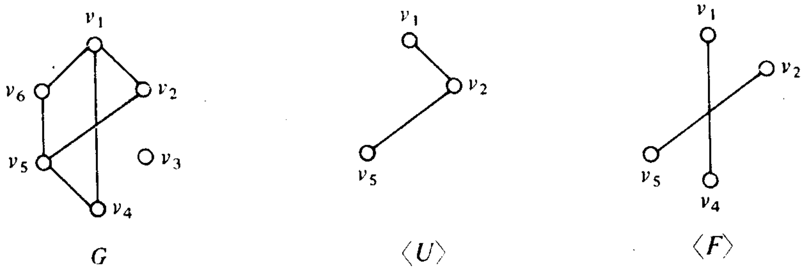


Figure 1.10 *Vertex-induced and edge-induced subgraphs*

The reader should be aware of possible confusion between nonisomorphic and nonidentical subgraphs. For example, in graph G_3 of Figure 1.6, how many subgraphs of G_3 have three vertices and three edges? The answer is obviously "two", since what is certainly desired here is the number of nonidentical such subgraphs. The reader could incorrectly give an answer of "one" here, interpreting the question as the number of nonisomorphic such subgraphs. Hence the reader must consider carefully the context in which the question is posed.

There are certain classes of graphs that occur so often that they deserve special mention and in some cases, special notation. We describe the most prominent of these in this section.

A graph G is *regular of degree r* if for each vertex v of G, deg $v = r$; such graphs are also called *r-regular*. The 3-regular graphs are referred to as *cubic* graphs. A graph is *complete* if every two of its vertices are adjacent. A complete (p, q) graph is therefore a regular graph of degree $p-1$ having $q = p(p-1)/2$; we denote this graph by K_p. In Figure 1.11 are shown all (nonisomorphic) regular graphs with $p = 4$, including the complete graph $G_3 \cong K_4$.

The *complement* $\bar{G}$ of a graph G is the graph with vertex set $V(G)$ such that two vertices are adjacent in $\bar{G}$ if and only if these vertices are not adjacent in G. Hence, if G is a (p, q) graph, then $\bar{G}$ is a $(p, \bar{q})$ graph, where $q + \bar{q} = \binom{p}{2}$. In Figure 1.11, the graphs G_0 and G_3 are complementary, as are G_1 and G_2. The complement $\bar{K}_p$ of the complete graph K_p has p vertices and no edges and is referred to as the *empty graph* of order p. A graph G is *self-complementary* if $G \cong \bar{G}$.

A graph G is *n-partite*, $n \geq 1$, if it is possible to partition $V(G)$ into n subsets $V_1, V_2, \ldots, V_n$ (called *partite sets*) such that every element of $E(G)$ joins a vertex of V_i to a vertex of V_j, $i \neq j$. If G is a 1-partite graph of order p, then $G \cong \bar{K}_p$. For $n = 2$, such graphs are called *bipartite graphs*; this class of

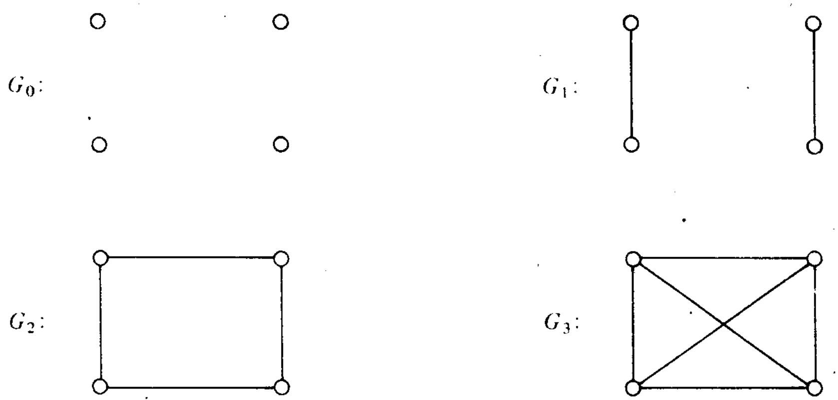


Figure 1.11 *The regular graphs of order 4*

graphs is particularly important and will be encountered many times. In Figure 1.12, a bipartite graph G_1 is shown; a second graph G_2, identical to G_1, is also given to emphasize the bipartite character of G_1. If G is a regular bipartite graph with partite sets V_1 and V_2, then $|V_1| = |V_2|$ (see Exercise 1.10; also see [ACLO1]).

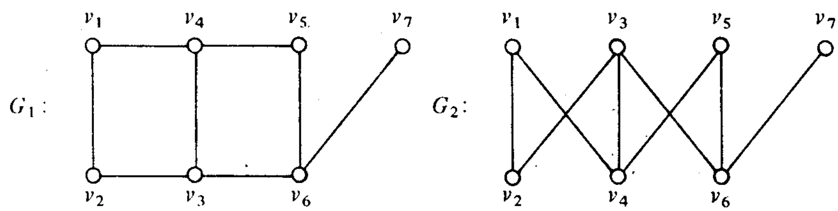


Figure 1.12 *A bipartite graph*

A *complete n-partite graph* G is an n-partite graph with partite sets V_1, V_2, $\ldots$, V_n having the added property that if $u \in V_i$ and $v \in V_j$, $i \neq j$, then $uv \in E(G)$. If $|V_i| = p_i$, then this graph is denoted by $K(p_1, p_2, \ldots, p_n)$. (The order of the numbers $p_1, p_2, \ldots, p_n$ is not important.) Note that a complete n-partite graph is complete if and only if $p_i = 1$ for all i, in which case it is K_n. If $p_i = t$ for all i, then the complete n-partite graph is regular and is also denoted by $K_{n(t)}$. Thus, $K_{n(1)} \cong K_n$.

A complete bipartite graph with partite sets V_1 and V_2, where $|V_1| = m$ and $|V_2| = n$, is then denoted by $K(m, n)$. The graph $K(1, n)$ is called a *star*.

There are many ways of combining graphs to produce new graphs. We next describe some binary operations defined on graphs. This discussion introduces notation that will prove very useful in giving examples. In the following definitions, we assume that G_1 and G_2 are two graphs with disjoint vertex sets.